

第四次上机作业（函数）求解思路

说明：

- 作业可用多种方法求解，可使用本文档思路，也可按自己的想法设计程序；
- 若程序中有逻辑问题（运行结果与预想不同），可使用“单步调试”查找具体出现逻辑问题的语句，详见“超星学习通”1.2节视频最后部分。

基础编程题

1. 编写函数，计算表达式 $s = \sum_{k=0}^n k!$ 的值($n \leq 10$)，形参n的值由主程序输入并传递，函数返回值为s。

提示：

(1) main函数中，由于n=10时，和值sum为最大值4037914，因此sum应定义为long型，以n为参数调用函数后，返回的long型结果的输出格式为%ld；

(2) 子函数中，由于 $0! = 1$ ，可设置和值sum的初始值为1，之后从1开始计算阶乘即可；

方法一：使用嵌套循环实现，外层循环实现从1到n的n次加和，内层循环实现求当前数的阶乘；

方法二：使用一层循环实现，实现从1到n的求阶乘及加和。原理： $i! = (i-1)! * i$ ，因此可利用上一个数的阶乘计算当前数的阶乘。

2. 编写数字加密和解密函数。测试值由主程序输入。

2.1 编写数字加密函数 `int encrypt(int n)`。其输入为一个四位数，返回为加密后的数。其加密方法为：

- 将该数每一位上的数字加9，然后除以10取余，作为该位上的新数字，
- 将第1位和第3位上的数字互换，第2位和第4位上的数字互换，组成加密后的新数。

例：输入 1257 输出 The encrypted number is 4601

提示：

(1) 在main函数中定义`int num1[4], num2[4]`；分别存放加密前数字、加密后数字的各位。以格式%1d利用循环输入num1的各位，再以num1, num2为实参调用encrypt函数（地址传递）；

(2) 在encrypt函数中，利用循环实现将num1数组中每个元素加9再对10取余，赋值给num2数组的元素。利用t按题意交换num2数组中元素值；

(3) 也可不用数组形式，定义`int num1, num2`，按第一次作业思路求解。

2.2 编写数字解密函数 `int decrypt(int n)`，其输入为一个使用2.1 encrypt 函数加密后的四位数，返回为解密后的数

例：输入 4601 输出 The decrypted number is 1257

提示：

(1) main函数中定义数组及元素输入与2.1类型；

(2) 在decrypt函数中，首先用t交换num2数组中元素值。num2中各元素为原始数据各位加9后对10取余的结果，因此利用循环对num2中各元素先加10再减9，赋值给num1数组的元素；

(3) 也可不用数组形式，定义`int num1, num2`。

3. 分别用函数和带参数的宏，实现从三个数中找出最大数。

提示：

使用带参数的宏时，注意宏体的参数要加()。

4.数组转置：在主函数中定义一个n（ $1 \leq n \leq 6$ ）维方阵，并从键盘读入数组元素；编写函数transpose(a, n)实现数组a转置；在主函数中输出转置后的方阵。

例：输入	4		输出	1	5	9	13
	1	2	3	4			
	5	6	7	8			
	9	10	11	12			
	13	14	15	16			
					2	6	10
					3	7	11
					4	8	12

提示：

(1) 可定义符号常量#define N 6, main函数中定义数组int a[N][N], 并从键盘输入n($1 \leq n \leq 6$), 即只使用数组a的前n行、前n列；

(2) transpose函数中, 转置可利用嵌套循环实现下标为i、j和j、i元素互换。注意循环时i、j的取值范围（仅对下三角元素转置即可, 否则上、下三角元素均转置仍得原矩阵）。

设计类编程题：

5. 在小学时我们就学习了分数的四则运算，即对两个分数进行加、减、乘、除等运算，要求用C语言编写4个分数的函数来实现四则运算。

输入：

分数1 操作符 分数2 (操作符为 + - * /)

输出：

计算结果

要求：

计算结果使用分数表示，并且为最简化。例如结果为2/6，则被简化为1/3

例：输入 1/6 + 1/3 输出 1/6+1/3=1/2

输入 1/6 - 1/3 输出 1/6-1/3=-1/6

提示：

(1) 程序包含如下函数：divisor函数求两个整数的最大公约数（具体思路可百度）、cal1函数实现分数加法、cal2函数实现分数减法、cal3函数实现分数乘法、cal4函数实现分数除法、main函数；

(2) 由于计算结果包含分子和分母，如 $2/3+3/5=19/15$ ，需cal1、cal2、cal3、cal4函数返回2个值，因此参数传递需使用“地址传递”实现一个函数“返回多个值”，即结果需用长度为2的数组存储（分子、分母），相应地，两个分数也需使用数组存储。

具体地，在main函数中定义int num1[2], num2[2], result[2]; 并定义char op; 从键盘输入的算式分别为num1中两个元素、op、num2中两个元素赋值。再用switch-case结构根据op值为'+、-、*、/'调用不同的函数进行计算，调用函数时实参为num1、num2、result（地址传递）；

(3) cal1函数实现分数加法：结果的分母先赋值为两个分数的分母相乘，再计算结果的分子（请自行思考对应表达式）。再调用divisor函数求结果分子、分母的最大公约数t，需注意求得的t可能为负，因此应以abs(分母)和abs(分子)为实参调用divisor函数（程序需包含math.h）。结果的分子和分母均除以t为最终结果。由于函数参数为地址传递，因此可通过形参数组改变实参数组result中的元素值；

(4) 其余3个函数类似。

6. 学生信息：编写4个函数实现下列功能（通过参数传递实现）：

- (1) 函数 `input()` ——输入 20 个学生姓名和高考总分；
- (2) 函数 `sort()` ——按高考总分从高到低的顺序排序，姓名顺序也随之调整；
- (3) 函数 `display()` ——显示所有学生姓名及其高考总分；
- (4) 函数 `search()` ——根据姓名用顺序查找方法找出该学生，查找成功返回学生下标，否则返回-1；
- (5) 主函数——①调用 `input()` 输入学生姓名及成绩；②调用函数 `sort()` 进行排序；③调用 `display()` 输入排序后的学生姓名及成绩；④输入一个姓名，调用函数 `search()`，根据返回值判定是否查找成功，若查找成功，输出该学生姓名及高考总分，否则显示查找失败。

提示：

- (1) `main`函数中，定义二维字符数组`name`存放学生姓名、一维`float`型数组`s`存放成绩；
- (2) `input`函数中，形参为数组`name`和`s`，利用循环实现对两个数组的输入；
- (3) `sort`函数中，形参为数组`name`和`s`，根据分数使用简单选择法或冒泡法对两个数组排序，注意字符串处理的特殊性（赋值时使用`strcpy`函数）；
- (4) `display`函数中，形参为数组`name`和`s`，利用循环实现二个数组的输出；
- (5) `search`函数中，形参为数组`name`和存放待查找学生姓名的一维字符数组`student`，进行`student`与`name`中各字符串的比较，若找到返回其下标，否则返回-1。

7. 字符串处理

定义两个 C 源程序文件：`fmain.c` 和 `fstring.c`。`fmain.c` 中包含主函数，实现字符串处理功能选项和字符串的输入与结果的输出。`fstring.c` 中包含两个函数 `delchar(s,c)`和 `strreverse (s)`，实现删除字符和字符串反转功能。

`delchar(s,c)`：在主函数中输入字符串和要删除的字符，此函数将字符串 `s` 中出现的所有 `c` 字符删除；再在主函数中输出删除后的字符串。

`strreverse (s)`：在主函数中读入字符串，此函数实现将字符串 `s` 反转；再在主函数中输出反转后的字符串。

SAMPLE OUTPUT:

```

1. Reverse a string.
2. Delete a character in a string.
Please input your choice: 1
Enter a string: holiday
Reversed string is: yadiloh
Would you like to continue? (y/n) Y
1. Reverse a string.
2. Delete a character in a string.
Please input your choice: 2
Enter a string: one world, one dream
Enter a character you want to delete: o
Deleted string is: ne wrld, ne dream
Would you like to continue? (y/n) N

```

提示：(本题不需要创建工程，利用文件包含实现)

(1) `fstring.c` 文件：仅包括如下 2 个函数

字符串反转函数 `stringreverse(s)`——将第 1 个字符与最后一个字符交换，第 2 个字符与倒数第 2 个字符交换，……，主要是找到字符串最后一个字符；

删除字符函数 `delchar(s,c)`——定义另一个一维字符数组 `temp`，在字符串 `s` 中从第 1 个字符到 `'\0'`，每次取一个字符与指定字符 `c` 比较，若不相同，则赋值给字符串 `temp`，最后将字符串 `temp` 拷贝给字符数组 `s`，用于“返回”；在原数组上修改也可以；

(2) `fmain.c` 文件：仅包括主函数

使用 `include` 命令包含 `fstring.c`，主函数中使用无限循环实现根据用户输入 (`y/n`) 决定是否结束程序执行。文本形式菜单使用 `printf` 语句打印即可，菜单选项使用 `switch-case` 即可实现，在不同的分支中调用不同的子函数实现相应功能。注意：在字符输入之前吸收回车。

思考题（选做）

8. N 个小朋友手拉手站成一个圆圈，从第一个小朋友开始报数，报到 6 的那个小朋友退出到圈外，然后他的下一位重新报“1”。这样继续下去，最后只剩下一个小朋友，他原来站在什么位置上呢？

编写一个函数 `counting()`，返回值是剩下的小朋友的位置，用数组记录小朋友退出圈外的顺序。

提示：本题为“约瑟夫环”问题，具体可百度。

(1) `main` 函数中可定义 `int kids[N]={0}`；该数组中元素为各小朋友的标志位，初始值均为 0 表示初始所有小朋友均未出圈。并定义 `int num[N]` 存放小朋友出圈的顺序，即 `num[0]` 存放第一个出圈小朋友的下标，……，`num[N-1]` 为最后一个出圈小朋友的下标。

`main` 函数以 `kids` 和 `num` 为实参调用 `counting` 函数（地址传递），函数执行结束后，`num[N-1]` 为剩下小朋友的下标，`num` 数组中的元素为小朋友出圈的顺序；

(2) `counting` 函数中，定义 `int i=0`；表示当前数到小朋友的下标，从第一个小朋友开始，因此初始值为 0。

设置控制变量为 `i_num` 的循环（`i_num` 从 0 到 $N-1$ ），实现每次找到出圈的小朋友，将其下标存储在 `num` 数组中下标为 `i_num` 的位置上。

循环体中每次均初始化 `count=0`（计数），当 `count<6` 时（循环）——

- ① 若 `kids[i]==0`，即下标为 `i` 的小朋友未出圈，则计数 `count++`。
- ② 若此时 `count==6`，则第 `i` 个小朋友出圈，设置其标志 `kids[i]=1`，并将其下标保存在 `num` 数组中 `num[i_num]=i`；
- ③ 检查下一个下标的小朋友（是否在圈中并计数），若 `i<N-1` 则 `i++`；否则，`i=0` 从第一个小朋友开始检查。